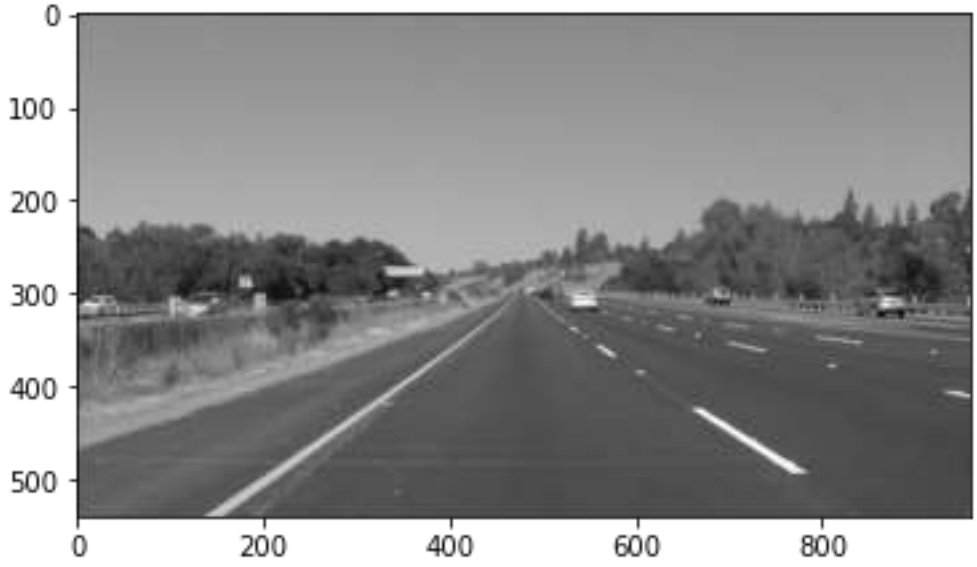
Reflection

Describe your pipeline. As part of the description, explain how you modified the draw_lines() function.

Given Image:

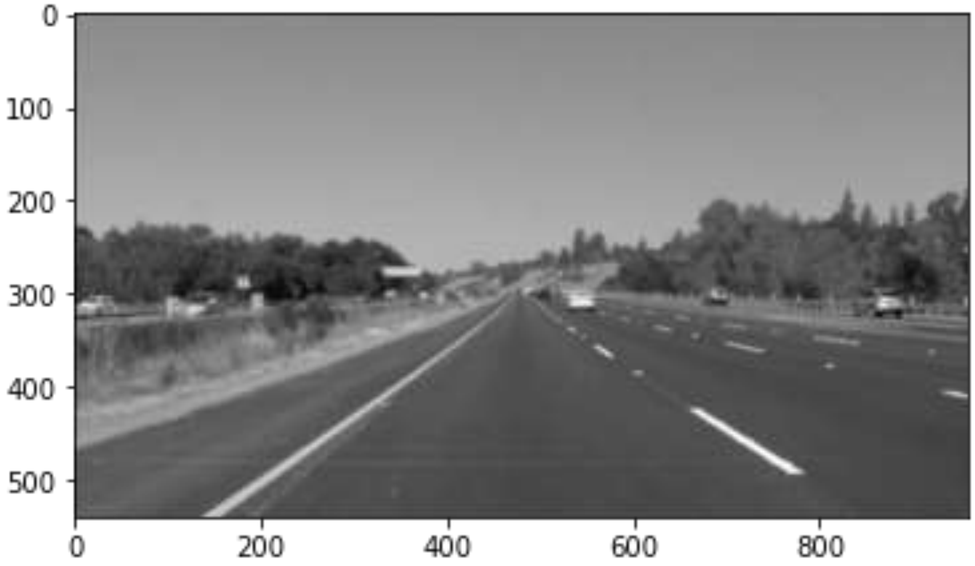
My pipeline consisted of the following steps:

i. Convert the given image into grayscale using the 'grayscale' function



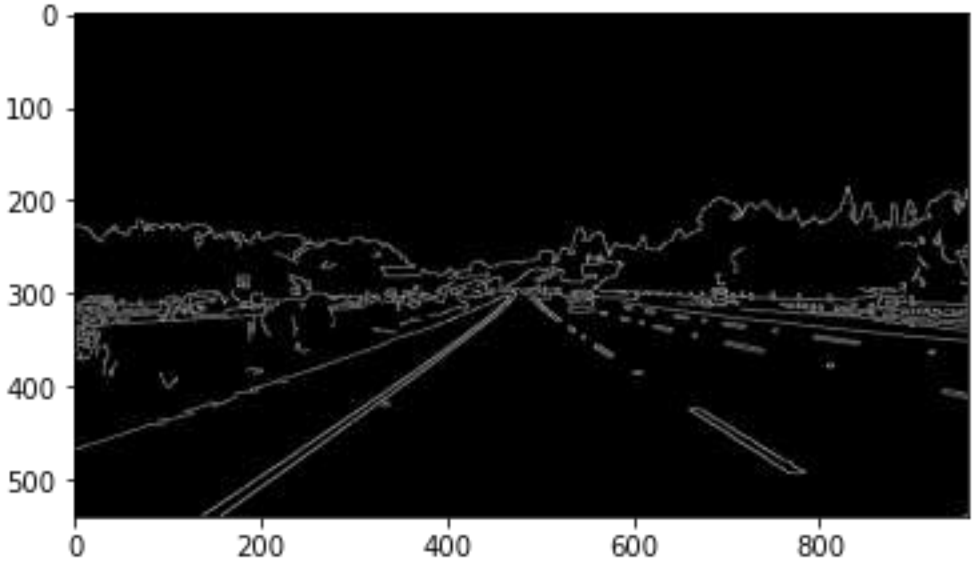
ii. Choose a kernel size and filter the gray image using the ‘gaussian_blur’ function

   a. Parameters: ‘kernel_size’

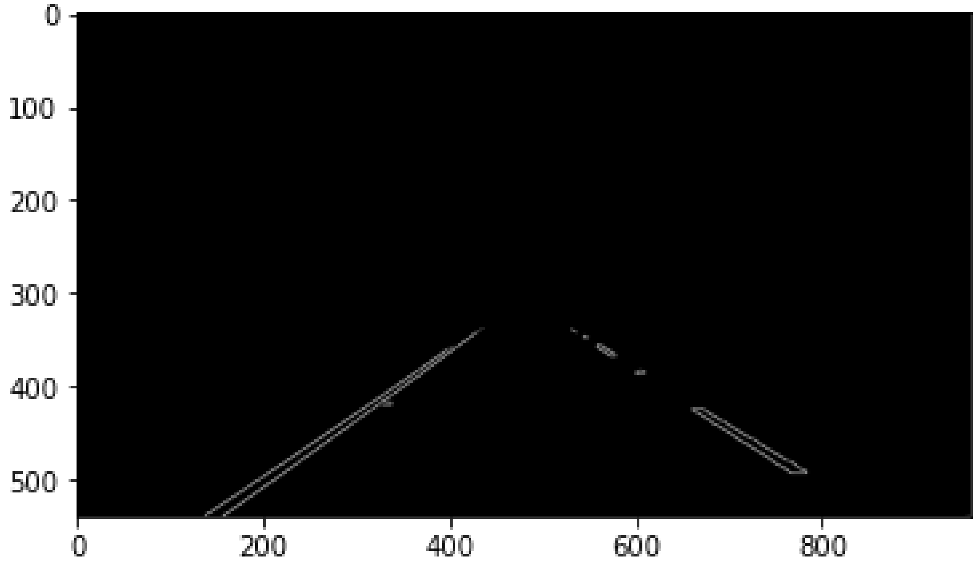


iii. Transform the filtered image into a line image using the ‘canny’ edge function. Canny edge method identifies huge differential change in pixel value to identify lines separating different objects in an image

   a. Parameters: ‘low_threshold’ and ‘high_threshold’ for the differential to identify an edge

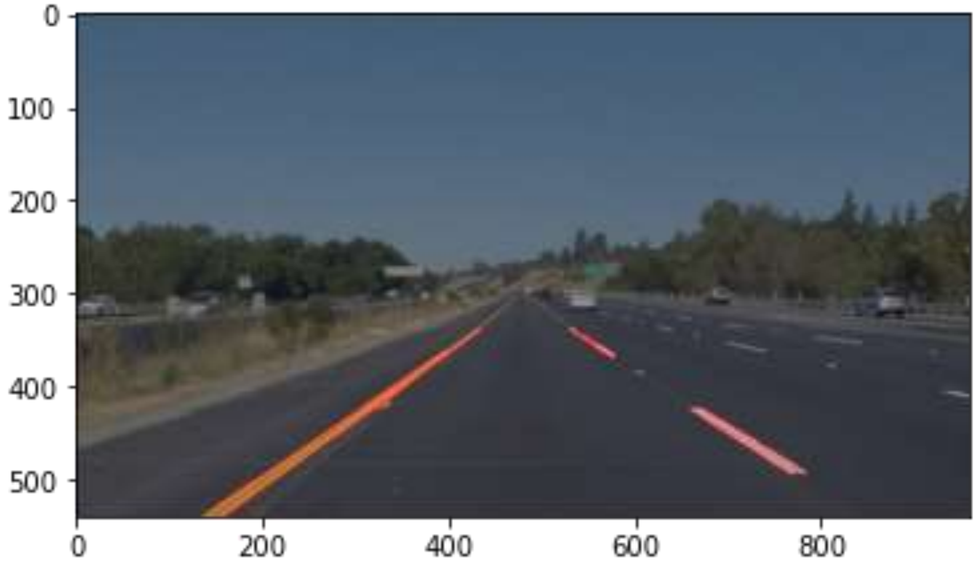


iv. The canny image was masked to carry over feature lines present in our area of interest and thus ignore everything else

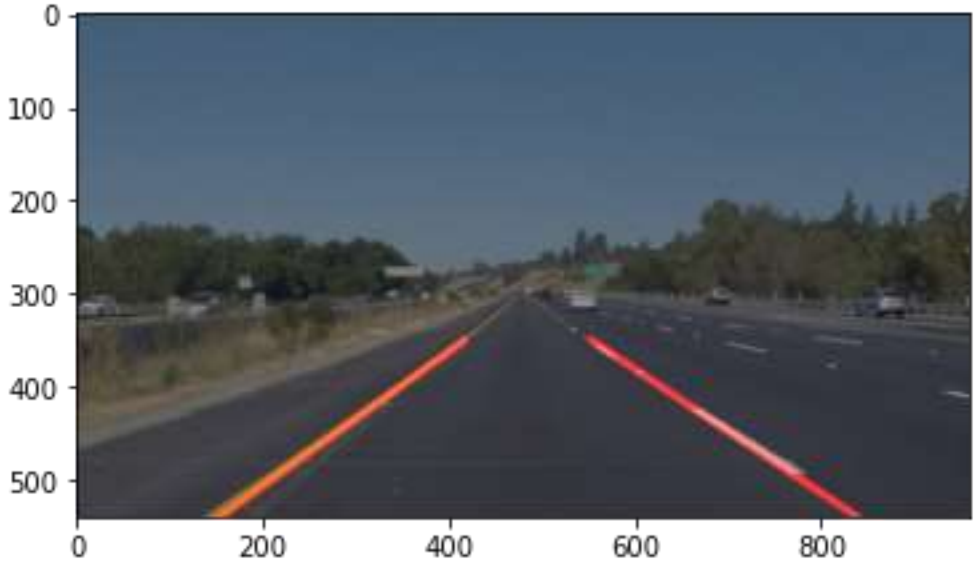


v. The canny image is transformed using Hough transformation which outputs the end points of all the important lines

a. Parameters: ‘rho’ is the distance of the grid in Hough space, ‘theta’ is the angular resolution, ‘threshold’ is the min number of intersections required for a line to be selected, ‘min_line_length’ is the minimum length of the line to make the cut and ‘min_line_gap’ is the maximum distance between segments which will be allowed to be connected.



vi. Finally, the draw_lines() was edited to give a single straight line for each side



Draw_lines()

These are the following steps/changes made to draw_lines():

i. Slope for each set of coordinates was calculated
ii. All lines were separated to belong to the left-side or right-side based on the slope. Similarly, the coordinates were also grouped based on the slope

iii. Average slope calculated for both sides
iv. Average intercept calculated for each side using the equation y = mx + c for each point
v. Max and min y-coordinate value for the area of interest were used to identify the corresponding x-coordinates for the start and the end of line using the mean slope and intercept calculated above
vi. These coordinates were used to plot the required line with cv2.line function

2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen the

vii. When lane would change abruptly
viii. When turning as the lanes will appear more like curves instead of straight lines
ix. When the orientation of the camera is changed
x. When the vehicle in front is very close to our vehicle

3. Suggest possible improvements to your pipeline

Possible improvements:

i. Merge the lines identified instead of just extrapolating
ii. Use a regression method to identify lines and let the algorithm identify the degree of these lines instead of assuming y = mx + c
iii. Masking can be based on regions containing only long lines
iv. Masking can help get rid of a very close vehicle from the canny image